UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/756,881	01/13/2004	XinLiang David Li	200313031-1	3438

22879 7590 03/20/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
IWASHKO, LEV	

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 03/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/756,881	LI ET AL.
	Examiner	Art Unit
	Lev I. Iwashko	2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7-17 and 20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *6,18 and 19* is/are allowed.

6)☒ Claim(s) *1-5,7-17 and 20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

1. The following are quotations of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

2. Claims 1-4, 10-13, and 20 are rejected under U.S.C. 102(e) as being anticipated by Grover et al. (US PG Pub 2003/0226133 A1)

Claim 1. A method for partitioning program modules, comprising:

- providing affinity weights among the modules; *(Section 0032, lines 6-8 – State the following: "The weight of an edge represents the affinity of two code blocks that are connected by the edge")*
- wherein a relationship between two modules constitutes an affinity weight for those two modules; *(Section 0032, lines 8-11 – State the following: "In one example, the affinity or weight of an edge between two code blocks may represent the number of times during a given period of execution that control flows from one of the code blocks to the other")*

- based on the affinity weights among the modules, providing a weight threshold; *(Section 0005, lines 16-17 – State the following: "The optimizer repeats the partitioning process until some threshold number of edges belong to a single partition")*
- and assigning a first module associated with an affinity weight that indicates the first module is most closely related to a second module; *(Column 0021, lines 7-10 – State the following: "Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types." Section 0030, lines 3-5 – State the following: "One table, a ranked edge list 241, includes a list of each edge in the weighted CFG 230 ranked by the assigned weight." Section 0032, lines 11-21 – State the following: "A code block may serve as a source code block for multiple edges. For example, code block G has edges that connect with both code block I and code block J. At code block G, the computer program may advance to code block I or code block J depending on whether one or more conditions in G are met. A code block may also serve as the destination block for one or more edges, such as code block K. Depending on the execution of the computer program upstream, code block K may be the destination block for code block I, code block J, or code block H")*
- and qualifying affinity weights that are associated with the first module, by comparing these affinity weights to the weight threshold; and assigning, to the group, all modules that are associated with the affinity weights qualified in the qualifying step. *(Section 0053, lines 1-16 – State the following: "If the weights of the edges are the same, procedure 900 goes to block 915 where the sizes of both edges are calculated. Again, as above, the size of each edge is the size of the source block plus the size of the destination block. After the weights of both edges are calculated, the procedure continues at decision block*

920 where a determination is made whether the size of the first edge Exy is greater than the second edge Epq. If so, procedure 900 goes to block 925 where the weight of Exy is recorded as greater than the weight of Epq. If not, procedure 900 goes to block 930 where the weight of Epq is recorded as greater than the weight of Exy. The procedure then returns. It will be appreciated that the case is unlikely but possible where the sizes of the two code blocks may be equal. In that case, some other tie-breaker may be used, or one code block may be arbitrarily selected as having a higher ranking")

Claim 2. The method of claim 1 wherein an affinity weight in the step of qualifying is qualified based on one or a combination of the following logical relationship with the weight threshold: equal to, greater than. *(Section 0053, lines 1-16 – State the following: "If the weights of the edges are the same, procedure 900 goes to block 915 where the sizes of both edges are calculated. Again, as above, the size of each edge is the size of the source block plus the size of the destination block. After the weights of both edges are calculated, the procedure continues at decision block 920 where a determination is made whether the size of the first edge Exy is greater than the second edge Epq. If so, procedure 900 goes to block 925 where the weight of Exy is recorded as greater than the weight of Epq. If not, procedure 900 goes to block 930 where the weight of Epq is recorded as greater than the weight of Exy. The procedure then returns. It will be appreciated that the case is unlikely but possible where the sizes of the two code blocks may be equal. In that case, some other tie-breaker may be used, or one code block may be arbitrarily selected as having a higher ranking")*

Claim 3. The method of claim 1 further comprising the steps of:

a) qualifying affinity weights that are associated with the modules assigned to the group by the step of assigning, by comparing these affinity weights to the threshold; and

b) assigning, to the group, all modules associated with the affinity weights qualified in step a).

Claim 4. The method of claim 1 wherein an affinity weight for two modules of the program 2 modules is provided based on one or more optimization opportunities between the two modules. *(Section 0037, lines 12-22 – State the following: "At block 515, G, EP, and SZ are input to the optimizer 210 (FIG. 2). At block 517, optionally (as indicated by the dashed line box), loop-back edges may be bias weighted to give them superior treatment in the ranked list of edges. Loop-back edges often have higher execution count than the execution count of edges within the loop. Bias weighting ensures that the most often executed transition between the loop tail and the loop head is given better locality in code space. An exemplary bias weighting procedure is illustrated later in the discussion of FIG. 8")*

Claim 10. The method of claim 1 being implemented as program instructions embodied in a computer-readable medium. *(Section 0008, lines 1-3 – State the following: "till another aspect of the invention is directed to a computer-readable medium encoded with computer-executable instructions")*

Claim 11. A method for partitioning modules, comprising:

a) providing a weight threshold; *(Section 0005, lines 16-17 – State the following: "The optimizer repeats the partitioning process until some threshold number of edges belong to a single partition")*

b) determining if there are modules remained to be partitioned, if there is not, then stopping the method; else proceeding to step c); *(Section 0005, lines 16-19 – State the following: "The optimizer repeats the partitioning process until some threshold number of edges belong to a single partition. In one embodiment, the partitioning process may be repeated until all edges belong to a single partition")*

c) finding among the modules that have not been assigned to a group a module associated with the highest affinity weight among the affinity

weights associated with the modules that have not been assigned to a group, and assigning this module to a new group; *(Section 0049, lines 6-14 – State the following: "Note that each partition represents a contiguous flow of control, and that the first partition tends to include the most weighted code blocks. Thus, if the code blocks are rearranged in an order consistent with the partitioning, the code blocks with the highest affinities should be located more closely, thus improving the working set and reducing paging. These partitions are then treated as atomic units and are further partitioned in another level, according to the operational flow diagram shown in FIG. 5")*

d) for each module in the new group created in step c) that has not been processed, identifying the each module as a first module; iterating through each module neighboring to the first module; wherein a first module neighboring to a second module if the first module and the second module is related by an affinity weight; if the neighboring module has not been assigned to a group, and an affinity weight between the neighboring module and the first module is qualified based on the weight threshold, then assigning the neighboring module to the new group; and e) proceeding to step b). *(Section 0050, lines 1-8 – State the following: "FIG. 8 is an operational flow diagram of an exemplary procedure 800 that implements bias weighting of loop back edges, as shown in block 517 of FIG. 5. The procedure starts at loop start block 810 where the loop begins. The loop repeats K times where K is the number of edges in a ranked edge list. When the loop has repeated for K times, procedure 800 returns. In each loop, the weight of a selected edge is increased if the edge is a loop back edge". Section 0051, lines 1-15 – State the following: "The loop starts at decision block 815 where whether the selected edge (En) is a loop back edge is determined. If not, the loop goes to loop end block 899 and returns to loop start block 810, increments, and repeats. Otherwise, if the edge (En) is a loop back edge, the loop continues at*

block 820 where the size of the edge is determined. The size of a code block relates to the amount of code (e.g., number of instructions) in that block. The size of the edge is the sum of the size of the source block and the destination block of the edge. After the size of the edge is determined, the loop then goes to block 825 where the weight of the edge is modified based on the size of the edge. In this embodiment, the new weight is the edge's old weight multiplied by the size of the edge. The loop then increments and repeats". Section 0052, lines 7-9 – State the following: "FIG. 9 is an operational flow diagram of an exemplary procedure 900 for ranking two edges that have the same weight. This procedure may be applied in conjunction with the creation of a ranked list of edges, as shown in block 415 of FIG. 4. The procedure 300 starts at decision block 910 where a determination of whether the weight of a first edge, shown as Exy, is the same as the weight of a second edge, shown as Epq. If not, then the edges are ranked based on their relative weights, and the procedure 900 returns". Section 0053, lines 1-16 – State the following: "If the weights of the edges are the same, procedure 900 goes to block 915 where the sizes of both edges are calculated. Again, as above, the size of each edge is the size of the source block plus the size of the destination block. After the weights of both edges are calculated, the procedure continues at decision block 920 where a determination is made whether the size of the first edge Exy is greater than the second edge Epq. If so, procedure 900 goes to block 925 where the weight of Exy is recorded as greater than the weight of Epq. If not, procedure 900 goes to block 930 where the weight of Epq is recorded as greater than the weight of Exy. The procedure then returns. It will be appreciated that the case is unlikely but possible where the sizes of the two code blocks may be equal. In that case, some other tie-breaker may be used, or one code block may be arbitrarily selected as having a higher ranking". Section 0054, lines 1-6 – State the following: "The above specification, examples and data provide

a complete description of the process and system of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended".)

Claim 12. The method of claim 11 wherein the affinity weight between the neighboring module and the first module is further qualified based on one or a combination of the following logical relationship: lesser than, equal to, greater than. *(Section 0053, lines 1-16 – State the following: "If the weights of the edges are the same, procedure 900 goes to block 915 where the sizes of both edges are calculated. Again, as above, the size of each edge is the size of the source block plus the size of the destination block. After the weights of both edges are calculated; the procedure continues at decision block 920 where a determination is made whether the size of the first edge Exy is greater than the second edge Epq. If so, procedure 900 goes to block 925 where the weight of Exy is recorded as greater than the weight of Epq. If not, procedure 900 goes to block 930 where the weight of Epq is recorded as greater than the weight of Exy. The procedure then returns. It will be appreciated that the case is unlikely but possible where the sizes of the two code blocks may be equal. In that case, some other tie-breaker may be used, or one code block may be arbitrarily selected as having a higher ranking")*

Claim 13. The method of claim 11 being implemented as program instructions embodied in a computer-readable medium. *(Section 0008, lines 1-3 – State the following: "till another aspect of the invention is directed to a computer-readable medium encoded with computer-executable instructions")*

Claim 20. A computer-readable medium embodying program instructions for performing a method for partitioning program modules, the method comprising:

a) providing affinity weights among the modules; *(Section 0032, lines 6-8 – State the following: "The weight of an edge represents the affinity of two code blocks that are connected by the edge")*

- wherein a relationship between two modules constitutes an affinity weight for those two modules; *(Section 0032, lines 8-11 – State the following: "In one example, the affinity or weight of an edge between two code blocks may represent the number of times during a given period of execution that control flows from one of the code blocks to the other")*

b) based on the affinity weights among the modules, providing a weight threshold; *(Section 0005, lines 16-17 – State the following: "The optimizer repeats the partitioning process until some threshold number of edges belong to a single partition")*

- and assigning a first module associated with an affinity weight that indicates the first module is most closely related to a second module; and *(Column 0021, lines 7-10 – State the following: "Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types." Section 0030, lines 3-5 – State the following: "One table, a ranked edge list 241, includes a list of each edge in the weighted CFG 230 ranked by the assigned weight." Section 0032, lines 11-21 – State the following: "A code block may serve as a source code block for multiple edges. For example, code block G has edges that connect with both code block I and code block J. At code block G, the computer program may advance to code block I or code block J depending on whether one or more conditions in G are met. A code block may also serve as the destination block for one or more edges, such as code block K. Depending on the execution of the computer program upstream, code block K may be the destination block for code block I, code block J, or code block H")*

c) qualifying affinity weights that are associated with the first module, by comparing these affinity weights to the weight threshold; and d) assigning, to the group, all modules that are associated with the affinity weights qualified in step c). e) qualifying affinity weights that are associated with all modules assigned to the group by step d), by comparing these affinity weights to the threshold; and f) assigning, to the group, all modules associated with the affinity weights qualified in step e). *(Section 0053, lines 1-16 – State the following: "If the weights of the edges are the same, procedure 900 goes to block 915 where the sizes of both edges are calculated. Again, as above, the size of each edge is the size of the source block plus the size of the destination block. After the weights of both edges are calculated, the procedure continues at decision block 920 where a determination is made whether the size of the first edge Exy is greater than the second edge Epq. If so, procedure 900 goes to block 925 where the weight of Exy is recorded as greater than the weight of Epq. If not, procedure 900 goes to block 930 where the weight of Epq is recorded as greater than the weight of Exy. The procedure then returns. It will be appreciated that the case is unlikely but possible where the sizes of the two code blocks may be equal. In that case, some other tie-breaker may be used, or one code block may be arbitrarily selected as having a higher ranking")*

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 5 is rejected under 35 U.S.C.103(a) as being unpatentable over Grover as applied to claim 1 above, further in view of Carini (US Patent 5,740,443) and Choi et al. (US Patent 6,179,491 B1).

Grover teaches the limitations of claim 1 for the reasons above.

Grover fails to teach the Claim 5, which states the following:

The method of claim 1 wherein the relationship between the two modules is based on one or a combination of:

- a number of calls across the two modules; a possibility for in-lining a function in a module of the two modules; *(Carini states the following: "Another approach to improve the effectiveness of automatic inline expansion relies on profiling information to select the call sites which should be inlined. See e.g., Pohua P. Chang, Scott A. Mahlke, William Y. Chen, and Wen mei W. Hwu, Profile-guided Automatic Inline Expansion for C Programs, Software--Practice and Experience, 22(5):349-369, May 1992. Profiling information can be used to obtain performance from autoinline mode which is equivalent to userinline mode but is less convenient for the user. The use of profiling information typically requires at least two compilations of the program, one compilation which is executed to generate the profiling information, and the other which is used to perform the automatic inlining. The performance improvement obtained with profiling based automatic inlining may also be sensitive to the input data set that was used to generate the profiling information" (Column 2, lines 46-61))*
- a characteristic of a call graph of functions in the two modules; *(Carini states the following: "After the complete program has been visited, (the model includes a provision for handling missing procedures), the interprocedural phase begins by constructing the program call graph (PCG). Each procedure is visited, the inlining inputs are collected, and an intermediate representation is generated, during the compilation phase of the separate compilation model. Cross file inlining and cloning, which may introduce cross file dependencies, are successfully handled by our compilation model. The interprocedural phase of our model is equivalent to the link phase of the separate compilation model. Procedures can be missing for various reasons, including the compilation of an incomplete program, calls to procedures written in another language, or calls to library procedures for which only the object code is available. The interprocedural phase*

detects missing procedures and reports them to the programmer" (Column 8, lines 8-22))

- a frequency of a global variable referenced in the two modules; *(Carini states the following: "Thus, a call site s is said to pass a variable X to a variable Y if and only if variable r is the same variable as X and is global to the called procedure, or X is passed-by reference to Y" (Column 5, lines 20-23))*
- a characteristic of a parameter passed between functions in the two modules; a possibility for de-virtualizing a virtual function in a module of the two modules; *(Choi states the following: "In addition, eliminating parts of a class hierarchy may enable the compilation process to perform other optimizations such as virtual-call elimination (i.e. replacing virtual method calls by direct method calls) and virtual inheritance elimination (i.e. replacing virtual inheritance by nonvirtual inheritance) that could previously not be applied" (Column 14, lines 53-58))*

It would have been obvious to one of ordinary skill in the art, having the teachings of the "System and Method for Improving a Working Set" of Grover, Carini's "Call-Site Specific Selective Automatic Inlining", and Choi's "Method and Apparatus for Slicing Class Hierarchies" before him at the time the invention was made, to combine the inventions to include the limitations presented in claim 5 so that the system would run more efficiently.

5. Claim 7 is rejected under 35 U.S.C.103(a) as being unpatentable over Grover as applied to claim 1 above, further in view of Broder et al. (US PG Pub 2004/0243554 A1)

Grover teaches the limitations of claim 1 for the reasons above.

Grover's invention differs from the claimed invention in that there is no specific reference to a weight threshold calculation.

Grover fails to teach claim 7, which states "The method of claim 1 wherein the weight threshold is calculated using a total value of the affinity weights among the modules." However, Broder states "FIG. 25 depicts the relationship of patterns with the WAND threshold, wherein a certain pattern is assigned a weight 2510, a second pattern is assigned a desired weight 2520,

until the last pattern is assigned a weight 2530. Collectively the assignments 2510, 2520, 2530 are used to produce a Threshold weight 2550. A summary of the use of the WAND technique 2800 is presented in FIG. 28. In FIG. 28, a first step involves initializing 2810, then evaluating the weighted sum of patterns 2820 and determining if the sum is above the threshold 2830. If the sum is below the threshold the pointers are advanced at step 2880 and the weighted sum of patterns evaluated again at step 2820. If the sum is above the threshold the method conducts a detailed evaluation at step 2840 and a determination at step 2850 if the value is above the minimum value in the heap (a heap of size n to keep track of the top n results, as discussed below). If not, control passes back to step 2880, otherwise the result is inserted into the heap at step 2860, the threshold and/or weights are modified at step 2870, and control passes back to step 2880" (Section 0256, lines 1-20).

It would have been obvious to one of ordinary skill in the art, having the teachings of Grover and Broder before him at the time the invention was made, to combine the inventions to calculate the weight threshold as taught by the "System and Method for Improving a Working Set" of Grover, and Broder's "System, Method and Computer Program Product for Performing Unstructured Information Management and Automatic Text Analysis", so that the system would be more accurate and the threshold would be based on actual values.

6. Claims 8-9, 14, and 16-17 are rejected under 35 U.S.C.103(a) as being unpatentable over Grover as applied to claims 1 and 7 above, further in view of Broder et al. (US PG Pub 2004/0243554 A1) and Schwartz et al. (US PG Pub 2004/0223941 A1).

Grover teaches the limitations of claim 1 and 7 for the reasons above.

Grover's invention differs from the claimed invention in that there is no specific reference to a percentages as they apply to thresholds.

Grover fails to teach claims 8-9, 14, and 16-17, which respectively state: "The method of claim 7 wherein the weight threshold is calculated using further a percentage value", "The method of claim 8 wherein the percentage value is derived from the capability of a compiler to handle a number of modules", "A method for providing an affinity weight threshold for use in partitioning program modules, comprising: providing a percentage value; providing affinity weights among the modules; providing a total value of the affinity weights; using the percentage value and the total value of the affinity weights to provide a percentage of the total of the affinity weights; using the percentage of the total of the affinity weights and a sum weight to provide the affinity weight threshold; the sum weight being the sum of at least two affinity weights", "The method of claim 14 wherein the affinity weight threshold is provided when an affinity weight added to the sum weight causing the sum weight being one or a combination of: equal to the percentage of the total of the affinity weights; and greater than the percentage of the total of the affinity weights", and "The method of claim 14 being implemented as program instructions stored in a computer-readable medium". However, Broder states the following: "FIG. 25 depicts the relationship of patterns with the WAND threshold, wherein a certain pattern is assigned a weight 2510, a second pattern is assigned a desired weight 2520, until the last pattern is assigned a weight 2530. Collectively the assignments 2510, 2520, 2530 are used to produce a Threshold weight 2550. A summary of the use of the WAND technique 2800 is presented in FIG. 28. In FIG. 28, a first step involves initializing 2810, then evaluating the weighted sum of patterns 2820 and determining if the sum is above the threshold 2830. If the sum is below the threshold the

pointers are advanced at step 2880 and the weighted sum of patterns evaluated again at step 2820. If the sum is above the threshold the method conducts a detailed evaluation at step 2840 and a determination at step 2850 if the value is above the minimum value in the heap (a heap of size n to keep track of the top n results, as discussed below). If not, control passes back to step 2880, otherwise the result is inserted into the heap at step 2860, the threshold and/or weights are modified at step 2870, and control passes back to step 2880" (Section 0256, lines 1-20). Grover also states the following: "The system comprises an optimizer that receives a compiled computer-executable program in binary format (binary code). After receiving the binary code, the optimizer generates a weighted control flow graph (CFG) and creates a ranked list of edges based on the information disclosed by the weighted CFG" (Section 0005, lines 3-7). Schwartz further states "A zinc-containing layered material with a solubility of less than 25% will have a measurable % soluble zinc value below a threshold value determined by the weight percent and molecular weight of the zinc compound. The theoretical threshold value can be calculated by the following equation: 0.25*wt. % Zn Compound in Composition*moles of Zinc in Compound*65.39(MW of Zn)/MW of Zn Compound" (Section 0014, lines 1-6).

It would have been obvious to one of ordinary skill in the art to combine the inventions to calculate the weight threshold as taught by the "System and Method for Improving a Working Set" of Grover, Broder's "System, Method and Computer Program Product for Performing Unstructured Information Management and Automatic Text Analysis", and Schwartz's "Composition comprising zinc-containing layered material with a high relative zinc lability" so that the system would be more efficient and the threshold would be based on actual percentage values.

7. Claim 15 is rejected under 35 U.S.C.103(a) as being unpatentable over Grover et al. as applied to claim 14 above, further in view of Schwartz et al. and Broder et al.

Grover, Schwartz and Broder teach the limitations of claim 14 for the reasons above.

Grover, Schwartz and Broder's inventions differ from the claimed invention in that there is no specific reference to making the percentage value tunable.

Grover, Schwartz and Broder fail to teach claim 15, which states "The method of claim 14 wherein the percentage value is tunable based on the capacity of a compiler." However, the fact that Grover, Schwartz and Broder do not mention that the percentage value is tunable is irrelevant, as making something adjustable does not change the purpose or functionality of the invention. Therefore, it would have been obvious to combine the "System and Method for Improving a Working Set" of Grover, Broder's "System, Method and Computer Program Product for Performing Unstructured Information Management and Automatic Text Analysis", and Schwartz's "Composition comprising zinc-containing layered material with a high relative zinc lability", so that the percentage would be adjustable allowing for increased user-friendliness.

For further information, please reference In re Stevens , 212 F.2d 197, 101 USPQ 284 (CCPA 1954) (Claims were directed to a handle for a fishing rod wherein the handle has a longitudinally adjustable finger hook, and the hand grip of the handle connects with the body portion by means of a universal joint. The court held that adjustability, where needed, is not a patentable advance, and because there was an art-recognized need for adjustment in a fishing rod, the substitution of a universal joint for the single pivot of the prior art would have been obvious.).

Allowable Subject Matter

8. Claims 6, and 18-19 are allowed.

The following is an examiner's statement of reasons for allowance:

Claim 6 declares that the affinity weight is calculated by "a formula f.sub.1w.sub.1+f.sub.2w.sub.- 2+ . . . f.sub.kw.sub.k; each weight w.sub.i is associated with a factor indicating a relationship between the two modules; and each f.sub.i is a weight percentage of the factor".

Claim 18 discloses "determining k factors; k being an integer number; each factor representing a distinct relationship between the two modules; and providing a sum of f.sub.iw.sub.i as the affinity weight; the subscript i running k times; wherein each w.sub.i is associated with a factor; each f.sub.i is a weight factor of a factor; and a sum of f.sub.i being equal to 100%."

Both Claims 6 and 18 disclose detailed formulas and variables that cannot be overcome by prior art.

Claim 19 is allowed, because it depends on Claim 18.

Tin conclusion, claims 6, and 18-19 are allowed due to their specific nature and because they all overcome obviousness.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lev I. Iwashko whose telephone number is (571)272-1658. The examiner can normally be reached on M-F (alternating Fridays), from 8-4PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lev Iwashko

MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100